RESEARCH ON COMMON SUBTREE MINING ALGORITHM IN DATA TREE WAREHOUSE

Nguyen Xuan Dung[a]
Vi Manh Hung[b]

[a]Faculty of Information Technology, Trung Vuong University

Email: nxdung@tv-uni.edu.vn

[b]Faculty of Information Technology, Trung Vuong University

Email: vmhung@tv-uni.edu.vn

Received: 28/01/2025; Reviewed: 24/02/2025; Revised: 08/3/2025; Accepted: 24/3/2025; Released: 30/3/2025

DOI: https://doi.org/.../.../...

ORCID iD: https://orcid.org/ 0009-0009-2877-0668

Important issue in mining database of trees is to find frequent occurs of subtrees. Since the number of frequent subtrees usually grows exponentially with the size of subtrees, therefore mining all frequent subtrees become infeasible for large tree size with traditional methods. There are several techniques which are used to prune the branches of enumeration tree that do not correspond to frequent subtrees. One of these techniques is heuristic method which is used to compute and identify frequent subtrees. In this paper, we present the canonical form for labelled rooted unordered trees the breadth-first canonical form (BFCF), then the canonical forms are applied to the frequent subtree mining problem.

Keywords: Graphs; The Breadth-First Canonical Form; Data mining; Frequent subtrees; Enumeration trees.

1. Giới thiệu

Dữ liệu lớn là một thuật ngữ dùng để mô tả các bộ dữ liệu có kích thước rất lớn, khả năng phát triển nhanh, khó thu thập, lưu trữ, quản lý và phân tích với các công cụ thống kê hay ứng dụng cơ sở dữ liệu truyền thống. Dữ liệu lớn rất quan trọng với các tổ chức, doanh nghiệp. Dữ liệu ngày một lớn và nhiều sẽ giúp các phân tích càng chính xác hơn. Việc phân tích chính xác này sẽ giúp doanh nghiệp đưa ra các quyết định giúp tăng hiệu quả sản xuất, giảm rủi ro và chi phí. Dữ liệu lớn cần đến các kĩ thuật khai thác thông tin rất đặc biệt do tính chất khổng lồ và phức tạp của nó. Dữ liệu lớn khác với dữ liệu truyền thống ở 4 điểm cơ bản như sau [12]:

• *Dữ liệu đa dạng hơn:* Khi khai thác dữ liệu truyền thống, ta thường phải trả lời các câu hỏi: Dữ liệu lấy ra kiểu gì, định dạng dữ liệu như thế nào nhưng đối với dữ liệu lớn ta không phải trả lời các câu hỏi trên. Hay nói cách khác là khi khai thác, phân tích dữ liệu lớn ta không cần quan tâm đến kiểu dữ liệu và định dạng của chúng, điều quan tâm là giá trị mà dữ liệu mang lại có đáp ứng được cho công việc hiện tại và tương lai hay không.

• *Lưu trữ dữ liệu lớn hơn:* Lưu trữ dữ liệu truyền thống vô cùng phức tạp và luôn đặt ra câu hỏi lưu như thế nào, dung lượng kho lưu trữ bao nhiêu là đủ, gắn kèm với câu hỏi đó là chi phí đầu tư tương ứng. Công nghệ lưu trữ dữ liệu lớn hiện nay đã phần nào có thể giải quyết được vấn đề trên nhờ những công nghệ lưu trữ đám mây, phân phối lưu trữ dữ liệu phân tán và có thể kết hợp các dữ liệu phân tán lại với nhau một cách chính xác và xử lý nhanh trong thời gian thực.

• *Truy vấn dữ liệu nhanh hơn:* Dữ liệu lớn được cập nhật liên tục, trong khi đó kho dữ liệu truyền thống không được cập nhật liên tục và trong tình trạng không theo dõi thường xuyên gây ra tình trạng lỗi cấu trúc truy vấn dẫn đến không tìm kiếm được thông tin đáp ứng theo yêu cầu.

• *Độ chính xác cao hơn:* Dữ liệu lớn khi đưa vào sử dụng thường được kiểm định lại dữ liệu với những điều kiện chặt chẽ, số lượng thông tin được kiểm tra thông thường rất lớn và đảm bảo về nguồn lấy dữ liệu không có sự tác động của con người vào thay đổi số liệu thu thập.

Dữ liệu lớn (Big Data) là thách thức rất lớn dẫn đến bùng nổ các phương pháp và kỹ thuật mới khai phá dữ liệu, một phần do kích thước thông tin rất lớn và một phần khác vì thông tin đa dạng, mở rộng hơn về các chủng loại cùng nội hàm của dữ liệu.

Nhiều năm nay, những thuật toán khai phá tập phổ biến đã được sử dụng để giải quyết nhiều vấn đề quan trọng trong khai phá dữ liệu nhằm phát hiện tri thức phục vụ cho việc trợ giúp ra quyết định cũng như phục vụ các hoạt động khác của con người. Những thuật toán hiệu quả cho việc tìm kiếm tập phổ biến cả tuần tự và song song trong những cơ sở dữ liệu rất lớn đã là một trong những thành công lớn

mang lại nhiều ý nghĩa trong lĩnh vực khai phá dữ liệu. Nhưng khi ứng dụng những kỹ thuật khai phá dữ liệu truyền thống vào những lĩnh vực phi truyền thống thì những phương pháp khai phá các tập mục phổ biến đang tồn tại không thật sự hiệu quả cũng như không thể mô hình được đầy đủ các yêu cầu đặt ra. Một phương pháp kết hợp trong việc mô hình các đối tượng khác nhau là sử dụng các đồ thị có nhãn, có hướng hoặc vô hướng để mô hình mỗi thực thể, hạng mục của đối tượng dữ liệu giúp cho việc giải quyết tốt vấn đề trên.

Trong bài báo này, trước tiên nhóm tác giả thực hiện nghiên cứu thuật toán hiệu quả để xác định chuỗi mã chuẩn theo chiều rộng BFSE của cây có thứ tự, sau đó nghiên cứu thuật toán lặp, không đệ quy để xác định chuỗi dạng chuẩn theo chiều rộng BFCF của cây không có thứ tự và làm cơ sở để nghiên cứu thuật toán khai phá hiệu quả các cây con phổ biến tiếp theo.

2. Tổng quan nghiên cứu vấn đề

Công việc phát triển các thuật toán khai phá cơ sở dữ liệu lớn các đồ thị là thử thách rất lớn, trong đó những bài toán chuyên sâu như tìm các đồ thị đẳng cấu, các đồ thị con phổ biến luôn đóng vai trò chính. Trong lĩnh vực khai phá dữ liệu đồ thị (Graph Mining) một lớp con đã rất quen thuộc được sử dụng rộng rãi là các cây và được ứng dụng trong nhiều lĩnh vực khác nhau như các tài liệu XML sử dụng cấu trúc cây để biểu diễn các phần tử - phần tử con và các mối quan hệ giữa thuộc tính - giá trị [5]. Trong khai phá truy cập Web, cây truy cập được sử dụng để biểu diễn các mẫu truy cập của những khách hàng khác nhau; trong phân tích sự tiến hóa của các phân tử, hay cây tiến hóa được sử dụng để mô tả lịch sử tiến hóa của các loài [1], [8]; trong mạng máy tính, cây được sử dụng để xác định vị trí của các gói tin [4], … Từ những ứng dụng nêu trên ta thấy, cây trong các ứng dụng thực tế thường là cây được gắn nhãn tại các đỉnh, nhưng nhãn của các cạnh không nhất thiết phải duy nhất [5], [6], [7].

Khi ta bắt đầu nghiên cứu các tập dữ liệu mới, thường ta chưa hiểu biết được các đặc trưng cơ bản của chúng. Chính các cấu trúc con phổ biến của tập dữ liệu thường hỗ trợ để ta hiểu và nghiên cứu sâu, chi tiết hơn về những dữ liệu đó. Wang và các cộng sự [2] đã áp dụng thuật toán khai phá cây con phổ biến vào cơ sở dữ liệu các phim ảnh trên Internet và đã phát hiện ra những cấu trúc chung của các tư liệu phim ảnh. Zaki và Aggarwal [5] đã giới thiệu một thuật toán để phân loại các tài liệu XML theo các cấu trúc con của chúng. Yun Chi và các cộng sự [3], [9], [10], [11] đã giới thiệu thuật toán đệ qui duyệt từ dưới lên để xác định dạng chuẩn theo chiều rộng BFCF (Breadth-First Canonical Form) hữu ích rất nhiều cho việc khai phá cơ sở dữ liệu các cây con phổ biến.

3. Cách tiếp cận và phương pháp nghiên cứu

Nhóm tác giả tập trung nghiên cứu và đánh giá tất cả các nguồn tài liệu, công trình khoa học trong nước và ngoài nước liên quan một cách hệ thống và toàn diện về cơ sở lý thuyết đồ thị, khai phá cơ sở dữ liệu các đồ thị và khai phá kho các cây dữ liệu.

3.1. Cơ sở về lý thuyết đồ thị

Như ta biết đồ thị được gắn nhãn G = (V, E, Σ, L) gồm tập các đỉnh V, tập cạnh E, bảng chữ Σ cho các nhãn của đỉnh, các cạnh và hàm gắn nhãn L: V E→ Σ. Đồ thị có hướng hoặc vô hướng nếu mỗi cạnh nối hai đỉnh là một cặp được sắp xếp hoặc không có thứ tự, không được sắp xếp tương ứng. Đồ thị liên thông nếu giữa hai đỉnh bất kỳ đều có ít nhất một đường đi nối giữa chúng, ngược lại là đồ thị không liên thông. Chu trình trong đồ thị là một đường đi mà đỉnh đầu và đỉnh cuối trùng nhau. Cây là một đồ thị vô hướng liên thông và phi chu trình. Cây có một đỉnh đặc biệt được gọi là gốc và thỏa mãn các tính chất sau:

- Mỗi đỉnh khác gốc đều có đúng một đỉnh vào;
- Có đúng một đường đi từ gốc tới mỗi đỉnh khác của cây.

Trong cấu trúc cây, đỉnh v trên đường đi từ gốc tới w được gọi là đỉnh "cha" của w, còn w được gọi là đỉnh "con" của v. Nếu w liền kề với v (có cạnh nối v với w) thì v là đỉnh "cha" của w, hay ngược lại w là đỉnh "con" của v. Kích thước của cây t được định nghĩa là số đỉnh của cây, ký hiệu là |t| = |V|. Ta qui ước cây có kích thước k sẽ được gọi là k-cây. Trong cây, một đỉnh v được gọi là "lá" nếu nó không có đỉnh "con". Bậc của mỗi đỉnh v là số đỉnh "con" của nó, ký hiệu là deg(v). Hiển nhiên, nếu v là lá thì deg(v) = 0. Chiều cao của cây t, ký hiệu là high(t), là độ dài của đường đi dài nhất trên cây (bắt đầu từ gốc). Thông thường, đối với k-cây thì độ dài của cây t luôn thỏa mãn high(t) < k. Một đỉnh trên cây được gọi là ở mức m nếu đường đi từ gốc tới nó có độ dài (số cạnh) là m, m ≤ h. Đỉnh gốc có mức là 0. Đỉnh không phải là gốc, không phải là lá được gọi là đỉnh trong của cây. Các cây có thể phân chia thành hai loại: cây có thứ tự và cây không có thứ tự. Một cây có thứ tự là cây trong đó các đỉnh "con" của mỗi đỉnh đều được xếp theo thứ tự từ trái qua phải, ngược lại được gọi là cây không có thứ tự.

Một cây t (với tập đỉnh Vt và tập cạnh Et) được gọi là cây "con" của cây s (với tập đỉnh Vs và tập cạnh Es) nếu và chỉ nếu thỏa mãn một là Vt ⊆Vs, hai là Et ⊆ Es và ba là các nhãn của các đỉnh trong Vt và các cạnh trong Et được bảo toàn trong cây s. Trong cây có thứ tự, các đỉnh "con" của một đỉnh bất kỳ đều được sắp xếp theo một thứ tự nhất định, ví dụ được liệt kê từ trái qua phải. Cây t và s được gọi là đẳng cấu với nhau (isomorphism) nếu tồn tại một ánh xạ 1-1 giữa hai tập đỉnh của t, s và bảo toàn được các nhãn của đỉnh và các cạnh. Một cây con t

có đẳng cấu trong cây s nếu tồn tại một đẳng cấu của t với một cây con của s.

Dễ dàng nhận thấy từ một cây không có thứ tự, ta có thể thấy, những cây có thứ tự đẳng cấu với nhau bằng cách thay đổi quan hệ giữa các đỉnh "con". Vì vậy, để nghiên cứu các cây không có thứ tự, ta cần xác định dạng chuẩn biểu diễn duy nhất cho các cây đó. Trường hợp đặc biệt đối với cây được gắn nhãn, hoàn toàn không mất tính tổng quát ta có thể giả thiết rằng tất cả các nhãn của các cạnh là đồng nhất, bởi vì mỗi cạnh đều được nối với một đỉnh "cha" của đỉnh đó. Ta có thể xem mỗi cạnh đó cùng với nhãn trên cạnh như là một phần được thể hiện trên nhãn của đỉnh "con". Đối với gốc không có cạnh đi tới nó, ta giả thiết đó là cạnh *null* (cạnh rỗng) nối với gốc. Từ đó, ta giả thiết rằng các nhãn trên các cạnh là tương đương, nghĩa là không cần gắn nhãn cho các cạnh mà chỉ cần xét các nhãn trên các đỉnh là đủ.

3.2. Dạng chuẩn của cây dữ liệu

Để biểu diễn cấu trúc một cây, ta có thể dựa vào các phương pháp duyệt cây. Có hai phương pháp duyệt cây là duyệt theo chiều rộng (breadth-first traversal) và duyệt theo chiều sâu (Depth-First Traversal). Yun Chi và các cộng sự [10, 11] đã giới thiệu thuật toán đệ qui duyệt từ dưới lên (Bottom-up) để xác định dạng chuẩn theo chiều rộng BFCF (Breadth-First Canonical Form) rất hữu ích cho việc khai phá các cây con phổ biến với độ phức tạp tính toán O(k2*d*log d), trong đó k là số đỉnh của cây và d là bậc cực đại của các đỉnh trong t (d = max{deg(v), v ϵ Vt).

Trong bài báo này, nhóm tác giả nghiên cứu thuật toán xác định mã chuỗi theo chiều rộng BFSE của cây có thứ tự và nghiên cứu thuật toán lặp để xác định dạng chuẩn BFCF của cây không có thứ tự. Thuật toán nhóm tác giả nghiên cứu ở đây có độ phức tạp thời gian tính toán là O(h*d*log d), trong đó h là chiều cao của cây và d là bậc cực đại của các đỉnh trong t (d = max{deg(v), v ϵ Vt), chiều cao h < k (số đỉnh của cây).

Không mất tính tổng quát, ta có thể giả thiết rằng có hai ký hiệu đặc biệt '$', '%' không nằm trong bảng chữ cái Σ được sử dụng để gắn nhãn cho các đỉnh của cây. Trên tập các nhãn mở rộng, Σ' = Σ ∪ {$, %} ta giả thiết luôn có một quan hệ thứ tự toàn phần ≤, nghĩa là:

• Giữa các nhãn v, w trên các đỉnh của cây luôn có quan hệ v ≤ w hoặc w ≤ v

• v ≤ $ ≤ %, với ∀v ϵ Σ.

Ở đây ta nghiên cứu mở rộng khái niệm mã chuỗi theo chiều rộng BFSE (Breadth-First String Encoding) [10] của cây có thứ tự bằng cách sử dụng ký hiệu % để đánh dấu kết thúc việc liệt kê các đỉnh từ phải qua trái sau mỗi mức của cây giúp cho việc xử lý các đỉnh trên cùng mức nhanh và hiệu quả hơn. Như vậy, mã chuỗi theo chiều rộng của cây có thứ tự được xác định bởi dãy các nhãn được liệt kê từ trên (gốc) xuống đến các lá và từ trái qua phải. Ở mỗi mức của cây, ta liệt kê các nhãn của cùng một cây "con" và bổ sung vào cuối dãy ký hiệu $, nếu đỉnh ở mức trước là lá (không có các đỉnh "con") thì cũng bổ sung $ và sau mỗi mức (sau đỉnh cuối cùng bên phải của mức) thì bổ sung ký hiệu % vào mã chuỗi các nhãn. Mỗi mã chuỗi BFSE sẽ biểu diễn tương ứng cho một cây có thức tự. Việc sử dụng ký hiệu % để đánh dấu kết thúc chuỗi các nhãn của các đỉnh trên cùng mức sẽ giúp cho việc xử lý các mã chuỗi theo chiều rộng nhanh và trực quan.

4. Kết quả nghiên cứu

4.1. Thuật toán BFSE xác định mã chuỗi theo chiều rộng của cây t có thứ tự

Dựa vào tư tưởng nêu trên, nhóm tác giả nghiên cứu thuật toán xác định mã chuỗi theo chiều rộng của cây có thứ tự như sau.

OrderedTree: tập (kiểu dữ liệu) các cây có thứ tự và ◦ là phép ghép hai chuỗi ký tự (nhãn).

Input: Cây có thứ tự t

Output: BFSE của t

```
Procedure BFSE(OrderedTree t)
{
    BFSE = L(root(t)) ◦ %;
    for j = 1 to high(t) do
            if( deg(v) == 0)  then
            BFSE = BFSE ◦ $;
            else
            for i = 1 to d do
            BFSE ◦ L(vi);
            BFSE = BFSE ◦ $;
            BFSE = BFSE ◦ %;
}
```

Để đánh giá độ phức tạp thuật toán, ta nhận thấy số vòng lặp của chu trình for ngoài bằng đúng chiều cao của cây, h = hight(t) và số vòng lặp của chu trình for bên trong nhỏ hơn hoặc bằng d = max{deg(v), v ϵ Vt). Do vậy, độ phức tạp của thuật toán 1 là ɵ(d*h), trong đó d bậc cực đại của các đỉnh trong t và h là chiều cao của cây.

4.2. Thuật toán xác định dạng chuẩn BFCF của cây không có thứ tự

Dựa vào quan hệ thứ tự toàn phần ≤ trên Σ' và việc xác định duy nhất mã chuỗi theo chiều rộng của cây có thứ tự, nhóm tác giả nghiên cứu thuật toán BFCF xác định dạng chuẩn BFCF của các cây

không có thứ tự bằng cách sắp xếp lại dãy nhãn của các đỉnh theo thứ tự ≤ UOrderedTree: tập các cây không có thứ tự.

Input: Cây t không có thứ tự

Ouput: Dạng chuẩn theo chiều rộng BFCF của cây t.

```
Procedure BFCF(UOrderedTree t)
    {
UOrderedTree t1 = t;
for i = high(t1) downto 1 do
   if(v1 ≼t v2 && L(v2) < L(v1)) then
      Đổi vị trí của 2 cây con v1 và v2;
    else if(v1 ≼t v2  && L(v2) == L(v1) &&
BFS(v2) < BFS(v1)) then
      Đổi vị trí của 2 cây con v1 và v2;
    }
```

Trong đó, BFS(v) là hàm xác định chuỗi các nhãn (kết thúc bằng $) của các đỉnh con của v đã được sắp xếp từ dưới lên. Ngoài ra, quan hệ L(v2) < L(v1), nghĩa là L(v2) ≤ L(v1) và L(v2) ≠ L(v1).

Thuật toán xác định BFCF của các cây không có thứ tự được thực hiện từ dưới lên trên theo các mức giảm dần của cây và từ trái qua phải, bằng cách so sánh các nhãn của gốc các cây "con" và sắp xếp chúng theo thứ tự ≤ không giảm trên Σ'. Kết quả ta thu được BFCF theo yêu cầu. Như vậy, từ một tập các cây có thứ tự đẳng cấu với nhau, sử dụng thuật toán để chuyển về cây dạng chuẩn BFCF.

Đối với cây t, thuật toán BFCF thực hiện vòng for lặp h = hight(t) lần. Trong mỗi vòng lặp, thì việc sắp xếp lại các đỉnh con của v ϵ Vt theo thứ tự ≤ sẽ cần ɵ(d*log d) thời gian để thực hiện, với d là số bậc cực đại của các đỉnh trong t. Như vậy độ phức tạp của thuật toán BFCF là ɵ(h*d*log d), trong đó h là chiều cao của cây và d là số bậc cực đại của các đỉnh trong t (d = max{deg(v), v ϵ Vt).

4.3. Cây con phổ biến

Giả thiết D = {T1, T2, …, Tn} là kho cơ sở dữ liệu các cây giao tác dữ liệu được gắn nhãn không có thứ tự (gọi tắt là cây giao tác) và cho trước cây mẫu t. Ta nói cây t xuất hiện trong cây giao tác T ϵ D, nếu t đẳng cấu với ít nhất một cây "con" của T và được ký hiệu δt(T) = 1, ngược lại δt(T) = 0. Ta nói rằng T hỗ trợ mẫu t nếu δt(T) = 1 và định nghĩa độ hỗ trợ của mẫu t trong D là support(t) = ΣT ϵD δt(T). Một mẫu (cây dữ liệu) t được gọi là phổ biến nếu độ hỗ trợ của nó lớn hơn hoặc bằng độ hỗ trợ cực tiểu minsup cho trước. Bài toán khai phá phát hiện cây con phổ biến là tìm tất cả các mẫu cây con phổ biến trong kho các cây dữ liệu cho trước. Bài toán khai phá các cây con phổ biến tương tự như pha đầu tiên của bài toán khai phá luật kết hợp, đó là pha khai phá tập mục phổ biến. Có hai loại thuật toán được sử dụng để khai phá tập mục phổ biến. Loại thuật toán thứ nhất gộp tất cả các tập mục phổ biến vào một cấu trúc và tạo thành dàn liệt kê (Enumeration Lattice), như các thuật toán Apriori-like [1] duyệt cây theo từng mức để xác định các tập mục phổ biến. Các thuật toán loại thứ hai gộp tất cả các tập mục phổ biến tạo thành cấu trúc cây liệt kê (Enumeration Tree) và duyệt theo từng mức của cây, ví dụ thuật toán duyệt cây theo chiều thẳng đứng của Agarwal và các cộng sự [5], [9]. Đối với các thuật toán Apriori-like, để tìm được các tập (k+1)-phổ biến thì phải lưu lại tất cả các tập k-phổ biến, do vậy đối với những cơ sở dữ liệu lớn sẽ đòi hỏi một không gian nhớ khá lớn. Trong khi đó, đối với các thuật toán duyệt theo cây liệt kê theo chiều thẳng đứng (chiều sâu) [7], [8], [9], thì chỉ cần lưu lại cha của ứng viên (k+1)-phổ biến trong cây liệt kê cần phải lưu trong bộ nhớ. Tuy nhiên, vấn đề liệt kê các cây con ứng viên đến nay vẫn chưa được giải quyết triệt để.

4.4. Cây liệt kê

Cây liệt kê (Enumeration Tree) được sử dụng để khai phá, phát hiện cây con phổ biến, đó là cây được xây dựng trên cơ sở liệt kê tất cả các cây con phổ biến của một kho các cây dữ liệu. Trong thuật toán của Asai và các cộng sự [6], [7] mỗi cây ứng viên sẽ tạo ra cây "con" duy nhất sau khi loại bỏ dư thừa trong cây liệt kê. Cây "con" được xác định duy nhất bằng cách loại đi đỉnh bên phải nhất (Rightmost Vertex), đỉnh cuối cùng theo thứ tự duyệt theo chiều sâu của các cây có thứ tự. Tiếp sau đó, Yun Chi, Yirong Yang và Richard R. Muntz [9], [10], [11] đã đưa ra khái niệm dạng chuẩn (Canonical Form) cho cây không có thứ tự được gắn nhãn và dựa vào mã chuỗi theo chiều rộng BFSE (Breadth-First String Encoding) [5] để xây dựng thuật toán khai phá các cây con phổ biến. Trong thuật toán của Yun Chi và các cộng sự thì để xác định được đỉnh bổ sung trong quá trình liệt kê các cây con ứng viên thì phải tính toán lại lại thứ hạng của tất cả các đỉnh ở đường biên của các cây con ở mức trước và thường thì số các cây ứng viên liệt kê là khá lớn. Việc cần phải tính toán lại thứ hạng của các đỉnh để bổ sung và sau đó thực hiện kiểm tra tính phổ biến của các cây con trong quá trình liệt kê là khá phức tạp và tốn kém thời gian, do vậy thuật toán phát hiện cây con phổ biến là không quá hiệu quả.

5. Thảo luận

Đồ thị được sử dụng rộng rãi trong việc biểu diễn dữ liệu và mối quan hệ giữa chúng. Trong tất cả các đồ thị, một lớp con đã rất quen thuộc với nhiều người đó là các cây (Trees), được ứng dụng trong nhiều lĩnh vực khác nhau. Bài toán cần nghiên cứu một vấn đề quan trọng trong khai phá cơ sở dữ liệu

các cây được gắn nhãn đó là tìm cây con phổ biến trong các bài toán khó vì một số lý do:

• Thông tin chung rất nhiều, đa dạng và phức tạp thường thu được từ các nguồn dữ liệu khác nhau mà khi ta bắt đầu nghiên cứu những tập dữ liệu mới thì chưa biết được các đặc trưng của chúng.

• Số các cây con phổ biến thường tăng theo hàm mũ của kích cỡ của cây nên việc khai phá được tất cả các cây con phổ biến trong kho cơ sở dữ liệu các cây là bài toán khó.

6. Kết luận

Trong bài báo này, nhóm tác giả nghiên cứu dạng biểu diễn chuẩn của cây dữ liệu và các thuật toán xác định chuỗi mã chuẩn, thuật toán chuyển các cây không có thứ tự về dạng chuẩn theo chiều rộng BFCF. Dựa vào dạng chuẩn theo chiều rộng BFCF của cây dữ liệu, nhóm tác giả sẽ tiếp tục nghiên cứu thuật toán khai phá dữ liệu các cây dữ liệu không có thứ tự để phát hiện cây con phổ biến trong cơ sở dữ liệu các cây dữ liệu.

Tài liệu tham khảo

Agarwal RC, Aggarwal CC, Prasad VVV, *A tree projection algorithm for generation of frequent itemsets,* J. of Parallel Distributed and Computing 61(3): 350-371, 2001.

C. Wang, M. Hong, J. Pei, H. Zhou, W. Wang and B. Shi, *Efficient Pattern-Growth Methods for Frequent Tree Pattern Mining,* Proc. Eighth Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD ’04), 2004.

F. Luccio, A.M. Enriquez, P.O. Rieumont and L. Pagli, *Bottom-Up Subtree Isomorphism for Unordered Labeled Trees,* Techni-cal Report TR-04-13, Universita di Pisa, 2004.

J. Huan, W. Wang and J. Prins, *Efficient Mining of Frequent Subgraph in the Presence of Isomorphism,* Proc. Int’l Conf. Data Mining (ICDM ’03), 2003.

M.J. Zaki and C.C. Aggarwal, XRules: *An Effective StructuralClassifier for XML Data,* Proc. Int’ l Conf. Knowledge Discovery andData Mining (SIGKDD ’03), 2003.

T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Satamoto and S. Arikawa, *Efficient Substructure Discovery from Large Semi-Structured Data, Proc. Second SIAM Int’l Conf. Data* Mining, Apr. 2002.

T. Asai, H. Arimura, T. Uno and S. Nakano, *Discovering Frequent Substructures in Large Unordered Trees,* Proc. Sixth Int’l Conf. Discovery Science, Oct. 2003.

S. Nijssen and J.N. Kok, *Efficient Discovery of Frequent Unordered Trees,* Proc. Int’l Workshop Mining Graphs, Trees, andSequences, 2003.

Y. Chi, Y. Yang and R.R. Muntz, *HybridTreeMiner: An Efficient Algorithm for Mining Frequent Rooted Trees and Free TreesUsing Canonical Forms,* Proc. 16th Int’l Conf. Scientific and Statistical Database Management (SSDBM ’04), June 2004.

Y. Chi, Y. Yang, Y. Xia and R.R. Muntz, *CMTreeMiner: Mining Both Closed and Maximal Frequent Subtrees,* Proc. Eighth Pacific Asia Conf. Knowledge Discovery and Data Mining (PAKDD ’04), May 2004.

Yun Chi, Yirong Yang, Richard R. Muntz, *Canonical forms for labelled trees and their applications in frequent subtree mining,* Knowledge and Information Systems 8: 203–234, 2005.

Hoan, N.C., *Tổng quan về dữ liệu lớn (Big Data), Kỷ yếu hội thảo khoa học Thống kê nhà nước với dữ liệu lớn,* trang 9-15, 2015.

NGHIÊN CỨU VỀ THUẬT TOÁN KHAI PHÁ CÂY CON PHỔ BIẾN TRONG KHO CÁC CÂY DỮ LIỆU

Nguyễn Xuân Dũng[a]
Vi Mạnh Hùng[b]

[a]*Khoa Công nghệ thông tin, Trường Đại học Trưng Vương*

Email:nxdung@tv-uni.edu.vn

[b]*Khoa Công nghệ thông tin, Trường Đại học Trưng Vương*

Email: vmhung@tv-uni.edu.vn

Ngày nhận bài: 28/01/2025; Ngày phản biện: 24/02/2025; Ngày tác giả sửa: 08/3/2025; Ngày duyệt đăng: 24/3/2025; Ngày phát hành: 30/3/2025

DOI: https://doi.org/.../.../...

ORCID iD: https://orcid.org/ 0009-0009-2877-0668

Vấn đề quan trọng trong bài toán khai phá kho các cây dữ liệu là tìm sự xuất hiện của các cây con phổ biến. Do số lượng các cây con phổ biến tăng theo hàm mũ kích cỡ của các cây dữ liệu, vì thế các phương pháp khai phá tất cả các cây con phổ biến truyền thống không mang lại hiệu quả đối với những cây dữ liệu kích cỡ rất lớn. Có một số kỹ thuật được sử dụng để tỉa bỏ các nhánh của cây liệt kê mà chúng không phải là cây con phổ biến, trong đó phương pháp heuristic được áp dụng để tổ chức tính toán và khai phá tất cả các cây con phổ biến một cách hiệu quả. Trong bài báo này, nhóm tác giả thực hiện nghiên cứu thuật toán xác định dạng chuẩn theo chiều rộng BFCF (Breadth-First Canonical Form) của cây không có thứ tự và sử dụng dạng chuẩn của cây không có thứ tự để khai phá các cây con phổ biến trong kho các cây dữ liệu.

Từ khóa: *Đồ thị; Dạng chuẩn theo chiều rộng; Khai phá dữ liệu; Cây con phổ biến; Cây liệt kê.*